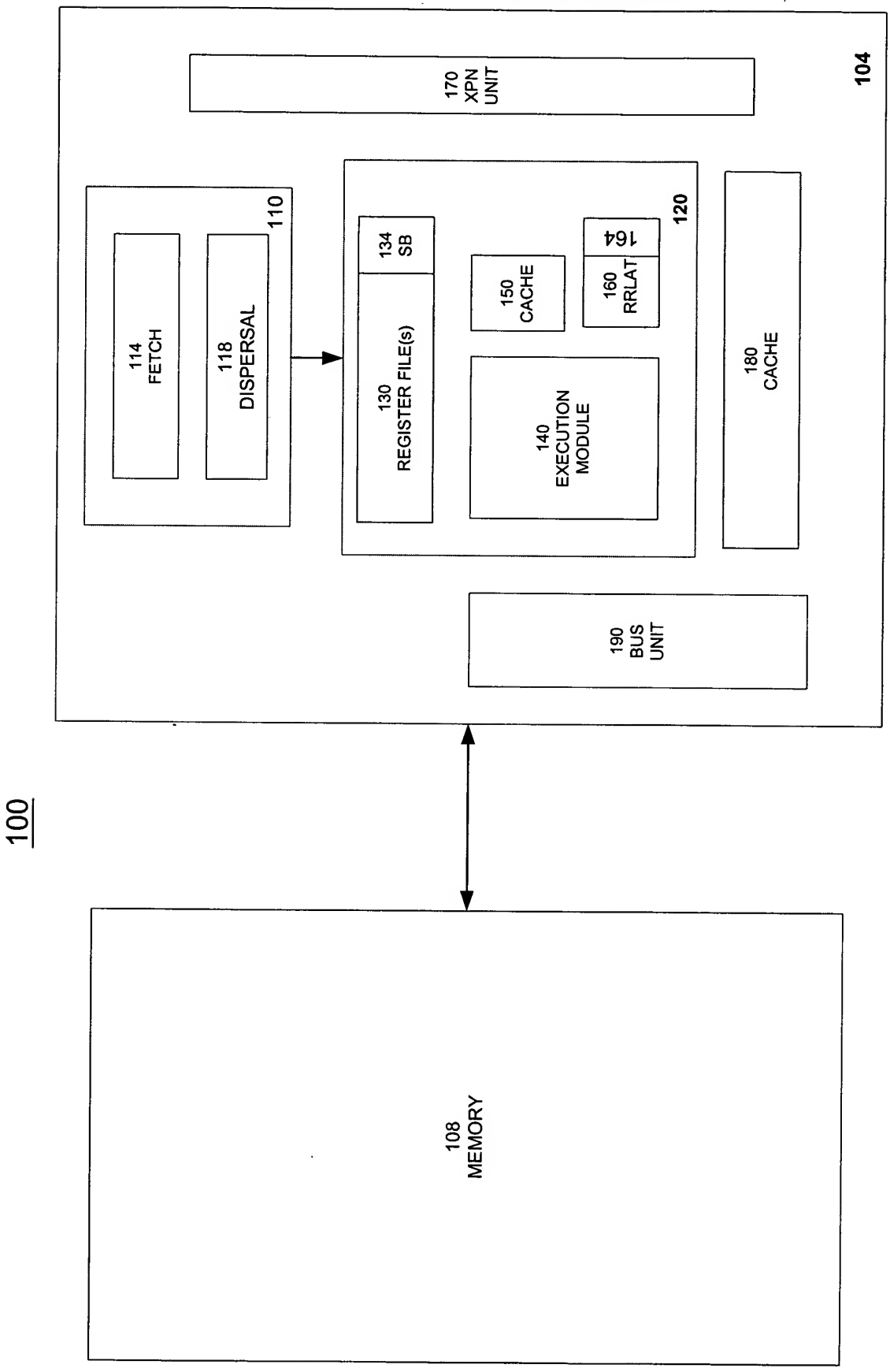
100
104
170 XPN UNIT
110
114 FETCH
118 DISPERSAL
120
134 SB
130 REGISTER FILE(s)
150 CACHE
164
160 RRLAT
140 EXECUTION MODULE
180 CACHE
190 BUS UNIT
108 MEMORY

Fig. 1

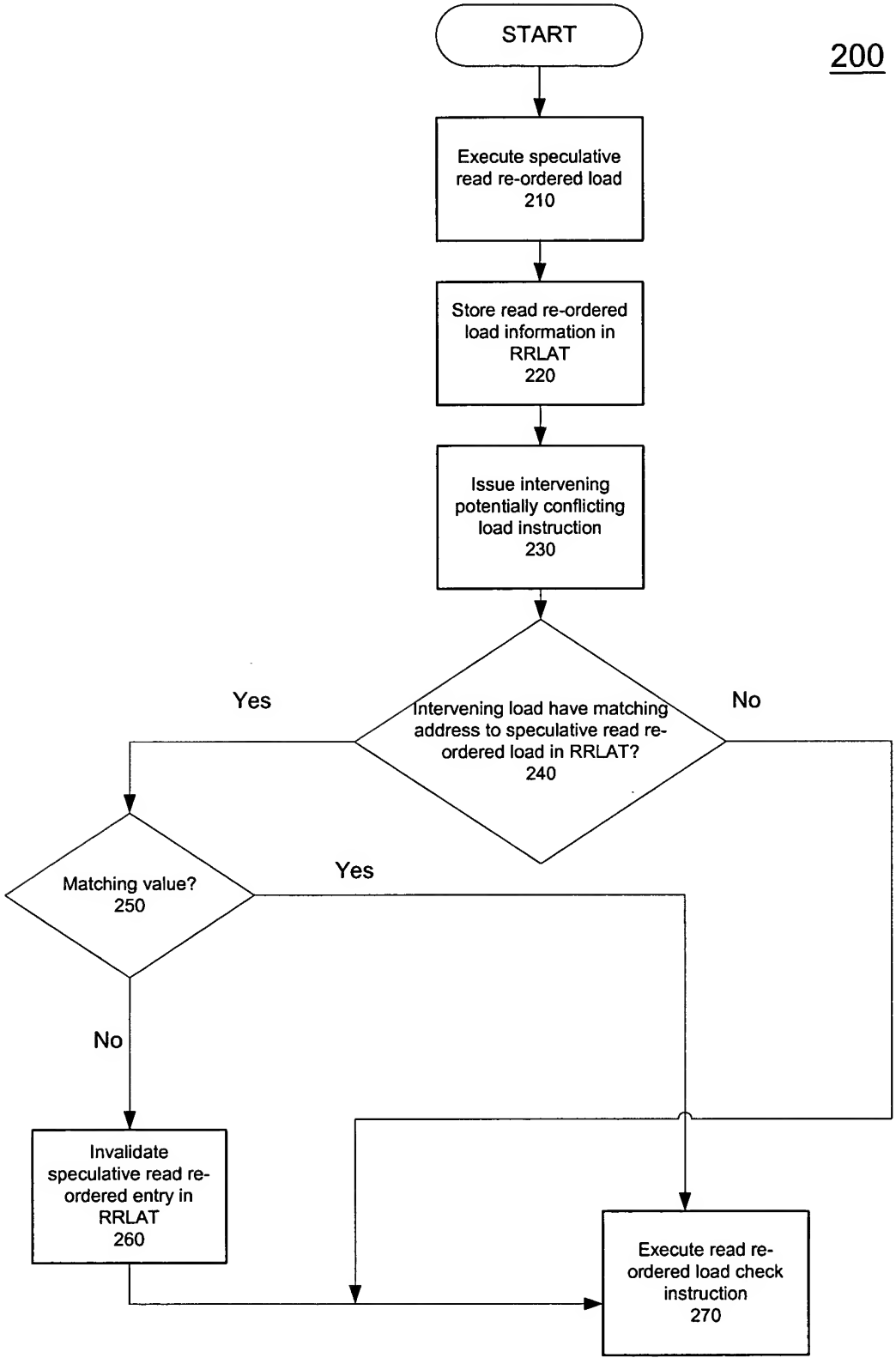
START
200
Execute speculative read re-ordered load 210
Store read re-ordered load information in RRLAT 220
Issue intervening potentially conflicting load instruction 230
Intervening load have matching address to speculative read re-ordered load in RRLAT? 240
Yes
No
Matching value? 250
Yes
No
Invalidate speculative read re-ordered entry in RRLAT 260
Execute read re-ordered load check instruction 270

Fig. 2

```
// p and q may refer to the same object;
// p.x initially 0;

t1 = ld [p.x];

...
// Another thread writes to q.x the value 1

t2 = ld [q.x];
t3 = ld [p.x];
```

Fig. 3a

```
// p and q may refer to the same object;
// p.x initially 0;

t1 = ld [p.x];

...
// Another thread writes to q.x the value 1

t2 = ld [q.x];
t3 = t1;
```

Fig. 3b

```
// p and q may refer to same object;
// p.x initially 0;

t1 = ld.rr [p.x]
// Another thread writes to q.x the value 1;

t2 = ld [q.x]
t3 = t1;

...
check.rr t1, recover
back:

...
recover:
t3 = ld [p.x]

...
br back
```

Fig. 3c

Physical RRLAT			
Target register ID	**Address**	**Value**	**Validity Bit**

Fig. 4